Key Feature Requirements - Query Results

Christopher Young
Sarah Martin
Thomas Jones

Key Feature 1

This query gets each flight, how many have booked it, the max capacity, and its scheduled arrival and departure timestamps

```
with plane_capacities as (
      select p.id as plane_id, sum(ptst.quantity) as plane_capacity
      from plane p
      inner join plane_type pt
      on (p.plane_type_id = pt.id)
      inner join plane_type_seat_type ptst
      on (ptst.plane_type_id = pt.id)
      group by p.id
),
passenger_counts as (
      select sf.id as scheduled_flight_id, sf.plane_id,
sf.departure_time, sf.arrival_time, count(*) as reservation_count
      from scheduled_flight sf
      inner join reservation r
      on (sf.id = r.scheduled_flight_id)
      group by sf.id, sf.plane_id, sf.departure_time, sf.arrival_time
)
select pcts.scheduled_flight_id,
      pc.plane_capacity,
      pcts.reservation_count,
      (pcts.reservation_count / pc.plane_capacity::decimal * 100) as
percent_booked,
      pcts.departure_time,
      pcts.arrival_time
from plane_capacities pc
inner join passenger_counts pcts
on (pc.plane_id = pcts.plane_id)
order by pcts.scheduled_flight_id asc
```

scheduled_flight_id	plane_capacity	reservation_count	percent_booked	departure_time	arrival_time
1	120	2	1.6666666667	2024-08-16 04:30:00.000	2024-08-16 06:35:00.000
2	120	1	0.8333333333	2024-08-16 06:50:00.000	2024-08-16 13:00:00.000
5	120	2	1.6666666667	2024-08-22 04:30:00.000	2024-08-22 06:30:00.000

6	120	1	0.8333333333	2024-08-16 07:30:00.000	2024-08-16 09:30:00.000
7	120	2	1.6666666667	2024-08-16 09:45:00.000	2024-08-16 12:00:00.000
8	120	1	0.8333333333	2024-08-16 12:15:00.000	2024-08-16 14:15:00.000
9	120	2	1.6666666667	2024-08-21 09:45:00.000	2024-08-21 11:45:00.000
10	120	133	110.8333333333	2024-08-23 00:00:00.000	2024-08-23 02:00:00.000
11	120	132	110	2024-08-23 02:00:00.000	2024-08-23 04:00:00.000
12	120	132	110	2024-08-23 04:00:00.000	2024-08-23 09:30:00.000
13	120	132	110	2024-08-23 09:30:00.000	2024-08-23 11:50:00.000
14	120	132	110	2024-08-23 11:50:00.000	2024-08-23 17:40:00.000
15	120	132	110	2024-08-23 17:40:00.000	2024-08-23 19:40:00.000
16	120	132	110	2024-08-23 19:40:00.000	2024-08-23 21:10:00.000
17	120	132	110	2024-08-23 00:00:00.000	2024-08-23 02:00:00.000
18	120	132	110	2024-08-23 02:00:00.000	2024-08-23 03:30:00.000
19	120	132	110	2024-08-23 03:30:00.000	2024-08-23 05:30:00.000
20	120	132	110	2024-08-23 05:30:00.000	2024-08-23 07:30:00.000
21	120	132	110	2024-08-23 07:30:00.000	2024-08-23 09:30:00.000
22	120	132	110	2024-08-23 09:30:00.000	2024-08-23 13:30:00.000

Key Feature 2
This query gets each flight, how many have booked it, the max capacity, and its scheduled arrival and departure timestamps.

```
with plane_capacities as (
      select p.id as plane_id, sum(ptst.quantity) as plane_capacity
      from plane p
      inner join plane_type pt
      on (p.plane_type_id = pt.id)
      inner join plane_type_seat_type ptst
      on (ptst.plane_type_id = pt.id)
      group by p.id
),
reservation_counts as (
      select sf.id as scheduled_flight_id, sf.plane_id,
sf.departure_time, sf.arrival_time, count(*) as reservation_count
      from scheduled_flight sf
      inner join reservation r
      on (sf.id = r.scheduled_flight_id)
      group by sf.id, sf.plane_id, sf.departure_time, sf.arrival_time
),
occupied as (
      select sf.id, count(*) as printed_boarding_pass
      from reservation r
      inner join seat s
      on (r.id = s.reservation_id)
      inner join scheduled_flight sf
      on (r.scheduled_flight_id = sf.id)
      where s.printed_boarding_pass_at is not null
      group by sf.id
)
select pcts.scheduled_flight_id,
      pc.plane_capacity,
      o.printed_boarding_pass,
      pcts.reservation_count
from plane_capacities pc
inner join reservation_counts pcts
on (pc.plane_id = pcts.plane_id)
inner join occupied o
on (pcts.scheduled_flight_id = o.id)
order by pcts.scheduled_flight_id asc
limit 22
```

scheduled_flight_id	plane_capacity	printed_boarding_pass	reservation_count
1	120	3	2
2	120	1	1
5	120	1	2
6	120	1	1
10	120	120	133
11	120	120	132
12	120	120	132
13	120	120	132
14	120	120	132
15	120	120	132
16	120	120	132
17	120	120	132
18	120	120	132
19	120	120	132
20	120	120	132
21	120	120	132
22	120	120	132
23	120	120	132
24	120	120	132
25	120	120	132
26	120	120	132
27	120	120	132

Key Feature 3: gets % seats sold, and total % passengers refunded

```
with total_seats as (
   select pt.plane_name,
          ptst.plane_type_id,
          sum(ptst.quantity) as num_seats
   from airline_booking2.plane_type_seat_type ptst
   inner join airline_booking2.seat_type st
       on (ptst.seat_type_id = st.id)
   inner join airline_booking2.plane_type pt
       on (ptst.plane_type_id = pt.id)
   group by pt.plane_name, ptst.plane_type_id
), seats_booked as (
   select p.plane_type_id,
          sf.id as flight_id,
          count(*) as num_booked
   from airline_booking2.seat s
   inner join airline_booking2.reservation r
       on (s.reservation_id = r.id)
   inner join airline_booking2.scheduled_flight sf
       on (r.scheduled_flight_id = sf.id)
   inner join airline_booking2.plane p
       on (sf.plane_id = p.id)
   where s.printed_boarding_pass_at is not null
   group by p.plane_type_id, sf.id
), overbooked_paid as (
   select sf.id as flight_id,
          count(*) as overbooked_paid_out
   from airline_booking2.reservation r
   inner join airline_booking2.scheduled_flight sf
       on (r.scheduled_flight_id = sf.id)
   inner join airline_booking2.payment pay
       on (r.id = pay.reservation_id and pay.amount < 0) -- Negative
payment indicates compensation
   group by sf.id
)
select

   coalesce(
       cast(sum(coalesce(sb.num_booked, 0)) as decimal(15,5))
       / cast(sum(ts.num_seats) as decimal(15,5)) * 100,
       0
   ) as percent_seats_sold,
   coalesce(
```

```
       cast(sum(coalesce(op.overbooked_paid_out, 0)) as
decimal(15,5))
       / cast(sum(coalesce(sb.num_booked, 0)) as decimal(15,5)) *
100,
       0
   ) as percent_passengers_refunded
from airline_booking2.scheduled_flight sf
inner join airline_booking2.plane p
   on (sf.plane_id = p.id)
inner join total_seats ts
   on (ts.plane_type_id = p.plane_type_id)
left join seats_booked sb
   on (sf.id = sb.flight_id)
left join overbooked_paid op
   on (sf.id = op.flight_id)
```

percent_seats_sold	percent_passengers_refunded
99.4794590735	0.0002739606

Key Feature 4:
Flight Performance Efficiency function
Calculates percentages based on the flights that have been canceled.

```
create or replace function flight_performance_efficiency() returns
table(percent_flights_on_time decimal(10,6), percent_flights_canceled
decimal(10,6)) as $$
      begin
      return query with flight_counts as (
      select
            count(*) as total_flights,
            sum(case
                        when fh.actual_departure_time is null and
fh.actual_arrival_time is null then 1
                        else 0
                  end
            ) as canceled_flights,
            sum(case
                        when fh.actual_departure_time is not null and
fh.actual_arrival_time is not null then 1
                        else 0
                  end
            ) as on_time_flights
      from airline_booking2.flight_history fh
      )
      select
            (coalesce(on_time_flights, 0) * 100.0 /
coalesce(total_flights, 1))::decimal(10,6) as
percent_flights_on_time,
            (coalesce(canceled_flights, 0) * 100.0 /
coalesce(total_flights, 1))::decimal(10,6) as
percent_flights_canceled
      from flight_counts;
      end;
$$ language plpgsql;

select * from flight_performance_efficiency()
```

percent_flights_on_time	percent_flights_canceled
99.745935	0.254065

Key Feature 5

Flight Revenue Estimation Query/Function

Calculates the expected revenue within a 10 day interval after a given start date

```
create or replace function flight_revenue_estimate(startdate date)
returns table(start_date date, end_date date, revenue decimal(10,2))
as $$
begin
   return query
   select
       (select min(departure_time)::date
        from airline_booking2.scheduled_flight
        where departure_time >= flight_revenue_estimate.startdate) as
start_date,
       (startdate + interval '10 days')::date as end_date,
       sum(p.amount)::decimal(10,2) as revenue
   from airline_booking2.scheduled_flight sf
   inner join airline_booking2.reservation r
       on sf.id = r.scheduled_flight_id
   inner join airline_booking2.payment p
       on r.id = p.reservation_id
   where sf.departure_time >= flight_revenue_estimate.startdate
     and sf.arrival_time < (startdate + interval '10 days')
   group by start_date, end_date; -- Group by to return correct
aggregates
end;
$$ language plpgsql;

select * from flight_estimate('08-21-24')
```

start_date	end_date	revenue
2024-08-21	2024-08-31	16,674,768.82